

Calibration and External Force Sensing for Soft Robots using an RGB-D Camera

Zhongkai Zhang, Antoine Petit, Jeremie Dequidt, Christian Duriez

Abstract—Benefiting from the deformability of soft robots, calibration and force sensing for soft robots are possible using an external vision-based system, instead of embedded mechatronic force sensors. In this paper, we first propose a calibration method to calibrate both the sensor-robot coordinate system and the actuator inputs. This task is addressed through a sequential optimization problem for both variables. We also introduce an external force sensing system based on a real-time Finite Element (FE) model with the assumption of static configurations, and which consists of two steps: force location detection and force intensity computation. The algorithm that estimates force location relies on the segmentation of the point cloud acquired by an RGB-D camera. Then, the force intensities can be computed by solving an inverse quasi-static problem based on matching the FE model with the point cloud of the soft robot. As for validation, the proposed strategies for calibration and force sensing have been tested using a parallel soft robot driven by four cables.

I. INTRODUCTION

During the last decade, many works have been done for the design, modeling and control of soft robots [1]. However, calibration and force sensing for soft robots are still open and challenging research areas, especially for robots with complex structures. Although deformations makes modelling difficult for soft robots, their observation can provide information for calibration and force sensing, which this work attempts to demonstrate.

Robotic calibration can be a key requirement to increase the accuracy of position control and force sensing. Indeed, the robot model can be improved with calibrated sensor-robot system, geometric and material parameters. These items are coupled between each other and can be simultaneously calibrated. Compared to rigid robots, the calibration of soft robots is more complex due to the difficulty to model the deformation mechanics of a soft material. The actuators can also be calibrated using embedded force sensors or position encoders in a decoupled manner with respect to other parameters. However, the use of sensors is expensive and sometimes limited by the space. Therefore, calibration of a sensor-robot system and actuators, without using sensors embedded in the actuators, is investigated in this paper. By minimizing the difference between the observed positions of a set of markers on the real robot and their known positions on its FE model, the rigid sensor-robot transformation and then the inputs (forces) of the actuators can be calibrated.

This work was supported by the Region Hauts-de-France, the project COMOROS (ERDF funds).

The authors are with INRIA, University of Lille, France. Contact: {zhongkai.zhang, antoine.a.petit, jeremie.dequidt, christian.duriez}@inria.fr

The approach is general and can be employed for robots with complex configurations.

Using external visual observations as a force sensor is an appealing approach which could simplify the robot design and provide a solution for applications where the size and weight constraints have to be considered. Among potential applications, this external system would be especially useful for delicate manipulation tasks with soft grippers. Prior works focused on the external force computation for continuum robot with contacts at the tip [2], [3]. In this paper, we introduce a method to solve a much more challenging problem: external force sensing in terms of location and intensity at multiple contact points on the soft robot.

This paper thus proposes a cheap and general strategy for calibration and force sensing of soft robots using an RGB-D camera. Calibration and force sensing are achieved using a real-time Finite Element Method (FEM) which has been employed for position control of soft robots [4], [5]. In contrast to our previous work [6], the designed external force sensing solution is marker-free and the accuracy is improved. Two main contributions are described: (1) a sequential calibration method based on the FE model of the soft robot to calibrate the sensor-robot system and the initial actuator inputs, using several fiducial markers on the surface of the robot. (2) A marker-free external force sensing strategy is introduced. Using a set of predefined feature points, the locations of potentially multiple external forces exerted on the robot are estimated based on the segmentation of the point cloud data. The corresponding intensities are computed according to the deflection between the point cloud and the FE model at the locations of the feature points and along their normals, assuming quasi-static equilibrium.

This paper is organized as follows. In Section II, the related research on calibration and external force sensing is reviewed. The quasi-static model of the soft robot and the previous marker-based external force sensing technique are introduced in Section III. Section IV presents the method to calibrate the sensor-robot system and the initial actuator inputs. In Section V, the marker-free external force sensing strategy is presented with the description of both the location estimation and intensity computation methods. The experiments on a soft parallel robot is presented in Section VI to validate the proposed techniques. Finally, a conclusion and some future directions are provided in Section VII.

II. RELATED WORK

Calibration of rigid robots has been intensively investigated

and most of the strategies focus on calibrating the robot geometrical parameters [7] and the robot-sensor system [8]. A few research works can also be found for the calibration of deformable robots. In [9] and [10], the geometrical parameters are calibrated for a continuum robot and a soft gripper respectively based on a constant curvature model. Model-free calibration is investigated in [11] where a machine learning technique is employed to calibrate a soft motion sensor system. By minimizing the deflection between loading and without loading, the material parameters of a soft object can be calibrated in [12], [13].

Current researches on force sensing for soft objects can be divided into two major categories: direct sensing and indirect sensing. The first category employs embedded sensors to measure the strain directly by placing the sensors at the tip of robotic tools [14], [15], around the shaft in a helical layout [16], or along the arc length [17] of the robotic manipulator. The direct sensing approaches generally lead to more accurate estimation of external forces. However, mounting sensors onto or inside the manipulator is not only expensive but also limited by the size of the instrument and by the softness of the considered robot (as sensors have their own stiffness). The idea of indirect sensing is to estimate external forces using the soft robot or the soft object itself as a force sensor. The actuator forces are employed to estimate the end-effector forces for a continuum manipulator [2] and a parallel continuum robot [3]. There are works about deflection-based force sensing to estimate tip forces. The contact forces can be estimated based on shape detection [18], tip pose measurement [19] and shaft curvature [20].

The work mentioned above are only suitable for tip force estimation. Sensing multiple external forces along the entire material or robot is more challenging because of the estimation of both force intensities and locations. In [21], kinematic-based contact detection and localization methods for multi-segment continuum robots are presented. A general strategy for soft robots with complex configurations is proposed in [6] where both the location and the intensity can be estimated based on the deflection of markers. RGB-D cameras have been used to detect the location of external forces for rigid robots [22] and hand-object manipulation [23]. In [13], as a close work to ours, a single external force acting on a soft object is computed through the registration of an FE model on point cloud data.

III. BACKGROUND: QUASI-STATIC MODELING AND MARKER-BASED EXTERNAL FORCE SENSING

In this section, we first introduce the Finite Element based quasi-static model for soft robots. A previous marker-based external force sensing technique [6] is then briefly reviewed, as an inspiration for the proposed marker-free strategy.

A. Quasi-static modeling

The discrete-time quasi-static equation of a soft robot with arbitrary external loads is deduced from a real-time FE method. By ignoring dynamics, the quasi-static equilibrium function of the soft robot at the $(k+1)th$ time step is given by:

$$\mathbf{f}_{ext} + \mathbb{F}(\mathbf{x}) + \mathbf{H}_a^T \lambda_a + \mathbf{H}_f^T \lambda_f = 0 \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ (n is the number of degrees of freedom of all nodes) is the position vector of all the FEM nodes and depends on the configuration of the soft robots. \mathbf{f}_{ext} and $\mathbb{F}(\mathbf{x})$ represent respectively the external loads (like gravity) and the internal stiffness forces. $\lambda_a \in \mathbb{R}^a$ (a is the number of actuators) and $\lambda_f \in \mathbb{R}^f$ (f is the number of external forces) are actuator and external forces. $\mathbf{H}_a^T \in \mathbb{R}^{n \times a}$ and $\mathbf{H}_f^T \in \mathbb{R}^{n \times f}$ provide the directions of the forces on the nodes exerted by the actuators and the external forces.

In (1), the internal stiffness forces $\mathbb{F}(\mathbf{x})$, which are non-linear, can be expressed based on first-order Taylor expansion, as:

$$\mathbb{F}(\mathbf{x} + d\mathbf{x}) \approx \mathbb{F}(\mathbf{x}) + \mathbf{K}(\mathbf{x}) d\mathbf{x} \quad (2)$$

where the tangent stiffness matrix $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ is highly sparse and depends on the actual position of the nodes \mathbf{x} . $d\mathbf{x}$ is the displacement between consecutive configurations.

The equilibrium equation of a soft robot at each time step can be obtained by combining (1) and (2). To reduce its dimension and make real-time computation possible, the equation can be projected into the constraint space by multiplying each term in the equilibrium by both matrices \mathbf{H}_e and \mathbf{H}_a , where \mathbf{H}_e maps the whole set of nodes to a predefined subset, or end-effectors, on which the forces will be effectively computed [24]. In this work, end-effectors consist in a set of feature points defined on the surface of the robot, as described in section V. Then, position vectors of the end effector $\delta_e \in \mathbb{R}^{3e}$ (e is the number of effectors) and the actuators $\delta_a \in \mathbb{R}^a$ (a is the number of actuators) can be written respectively as:

$$\delta_e = \mathbf{W}_{ea}(\mathbf{x}) \lambda_a + \mathbf{W}_{ef}(\mathbf{x}) \lambda_f + \delta_e^{free} \quad (3)$$

$$\delta_a = \mathbf{W}_{aa}(\mathbf{x}) \lambda_a + \mathbf{W}_{af}(\mathbf{x}) \lambda_f + \delta_a^{free} \quad (4)$$

where the compliance matrices can be computed, based on the configuration of the soft robot, as

$$\mathbf{W}_{ij}(\mathbf{x}) = \mathbf{H}_i \mathbf{K}^{-1}(\mathbf{x}) \mathbf{H}_j^T \quad (5)$$

with i, j standing for the subscript e, a and a, f , respectively.

When the actuator forces and external forces are computed, the configuration of the soft robot is updated by

$$d\mathbf{x} = \mathbf{K}(\mathbf{x})^{-1} \mathbf{H}_a^T \lambda_a + \mathbf{K}(\mathbf{x})^{-1} \mathbf{H}_f^T \lambda_f + d\mathbf{x}^{free} \quad (6)$$

Using the compliance matrices, we can get a measure of the mechanical coupling between effectors and actuators ($\mathbf{W}_{ea} \in \mathbb{R}^{e \times a}$), effector and external forces ($\mathbf{W}_{ef} \in \mathbb{R}^{e \times f}$), actuators and actuators ($\mathbf{W}_{aa} \in \mathbb{R}^{a \times a}$), and actuators and external forces ($\mathbf{W}_{af} \in \mathbb{R}^{a \times f}$). If the actuators and external forces are applied on different nodes or with different directions, there will be no linear dependency between the lines of \mathbf{H}_a , \mathbf{H}_f and \mathbf{H}_e so that the matrix \mathbf{W}_{aa} is positive-definite and $\mathbf{W}_{ea}, \mathbf{W}_{ef}$ and \mathbf{W}_{af} has full row rank or full column rank.

B. Marker-based external force sensing

In this section, we briefly describe the strategy used in [6] for external force sensing based on the Finite Element Method and markers placed on the robot, which the method proposed here is inspired by. As shown in Fig. 1, the actuator input λ_a is sent to both the soft robot and its FE model, so the influence of the actuator inputs can be compensated. When an external force acts on the soft robot, there is a deflection between the FE model and the real robot. The external forces λ_f are computed to make the feature points on the FEM model δ_e align to their corresponding points on the real robot δ_e^r .

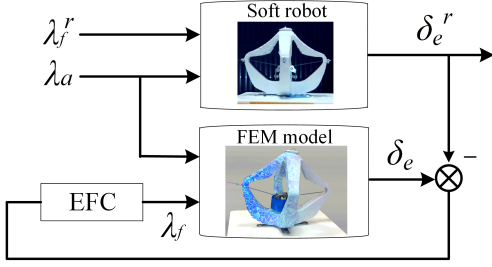


Fig. 1. Framework of external force sensing for soft robots. EFC is the algorithm of external force computation. λ_a is the actuator input for both the soft robot and its FE model. λ_f^r and δ_e^r are respectively the external force and position of effectors for the soft robot. λ_f and δ_e are the corresponding variables for FE model.

IV. CALIBRATION

Similarly to [6], the external force sensing method consists in exploiting the deviation between the real robot, observed with a vision sensor and its FE model. Therefore, calibrating the two systems is necessary to make sure that this deviation is minimized without external forces and are measured in the same coordinate system. To simplify the calibration method, we assume that both the geometric and material parameters of the soft robot are modeled accurately. The items to be calibrated are thus the rigid transformation between the robot and the vision sensor (the RGB-D camera), in an eye-to-hand manner, and the initial actuator inputs.

A set of feature points are set on the surface of the robot. As an offline procedure, we consider the use of fiducial markers placed on the surface. Their known positions in the robot frame, defined as δ_{robot} , are matched with their observed positions in the camera frame, defined as δ_{cam} . The goal of the calibration technique is then to minimize a position error with respect to the camera/robot rigid transformation (\mathbf{R}, \mathbf{T}) and the initial actuator input λ_a :

$$\min_{\mathbf{R}, \mathbf{T}, \lambda_a} \sum_{i=1}^n \|\mathbf{R} \delta_{cam}^i + \mathbf{T} - \delta_{robot}^i\|^2 \quad (7)$$

$$\text{s.t. } \lambda_a \in \Omega$$

with $\delta_{robot} = [\delta_{robot}^0 \dots \delta_{robot}^n]^T = \mathbf{W}_{ea} \lambda_a + \delta_e^{free}$. $\|\cdot\|_2$ is the 2-Norm and Ω is the constraint for the actuator input. n is the number of markers on the robot surface. $\mathbf{T} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ are respectively the translation vector and the rotation matrix which align each observed marker to their positions in the robot base frame defined for the FE model.

The method consists of two sequential steps: the computation of the optimal rigid transformation and the optimization of the actuator inputs.

The correspondences between two sets of points generate a linear least-squares problem that can be solved robustly using the Singular Value Decomposition (SVD) method. By first setting $\lambda_a = 0$, the optimal rigid transformation can be obtained by solving (7) with respect to \mathbf{T} and \mathbf{R} .

Based on the estimated optimal rigid transformation $(\hat{\mathbf{R}}, \hat{\mathbf{T}})$, the actuator input is then estimated by solving:

$$\min_{\lambda_a} \frac{1}{2} \|\mathbf{W}_{ea} \lambda_a + \delta_e^{free} - \delta_{tran}\|^2 \quad (8)$$

$$\text{s.t. } \lambda_a \in \Omega$$

with $\delta_{tran} = \hat{\mathbf{R}} \delta_{cam}^i + \hat{\mathbf{T}}$. It can be converted to a standard quadratic programming (QP) formulation:

$$\min_{\lambda_a} \frac{1}{2} \lambda_a^T \mathbf{Q} \lambda_a + \mathbf{c}^T \lambda_a \quad (9)$$

$$\text{s.t. } \lambda_a \in \Omega$$

where $\mathbf{Q} = \mathbf{W}_{ea}^T \mathbf{W}_{ea}$ and $\mathbf{c} = \mathbf{W}_{ea}^T (\delta_e^{free} - \delta_{tran})$.

The calibration algorithm is also reported in Algorithm 1. In (7), \mathbf{W}_e and δ_e^{free} cannot be pre-computed based on the initial configuration of the FE model. Therefore, after the computation of λ_a , the configuration of the FE model should be updated with I_2 iterations in order to compute more accurate values for \mathbf{W}_e and δ_e^{free} . The calibration stops after I_1 iterations.

Algorithm 1 Calibration algorithm

1. **Place** markers on the robot, with corresponding positions on the FE model
 2. **Read** the positions of markers from the vision sensor
 3. **While** $Iteration < I_1$
 - Compute** positions of the markers on the FE model frame with (3)
 - Compute** rigid transformation $(\hat{\mathbf{R}}, \hat{\mathbf{T}})$
 - Compute** positions of the markers in the camera frame
 - While** $Iteration < I_2$
 - Compute** actuator forces λ_a (9)
 - Update** the configuration of FE model with (6)
 4. **Output** \mathbf{T} , \mathbf{R} and λ_a
-

V. EXTERNAL FORCE SENSING

The objective of this section is to describe the method to sense the external forces based on the FE model and using the RGB-D camera. We can distinguish two parts: location recognition based on the segmentation of the point clouds provided by the camera, and intensity computation based on the deflection of a set of preliminarily selected feature points.

A. Location recognition and feature points matching

1) *Point cloud filtering*: The surfaces of both the soft robot and the surrounding objects (the manipulation tools or an obstacle for instance) are captured by the RGB-D camera.

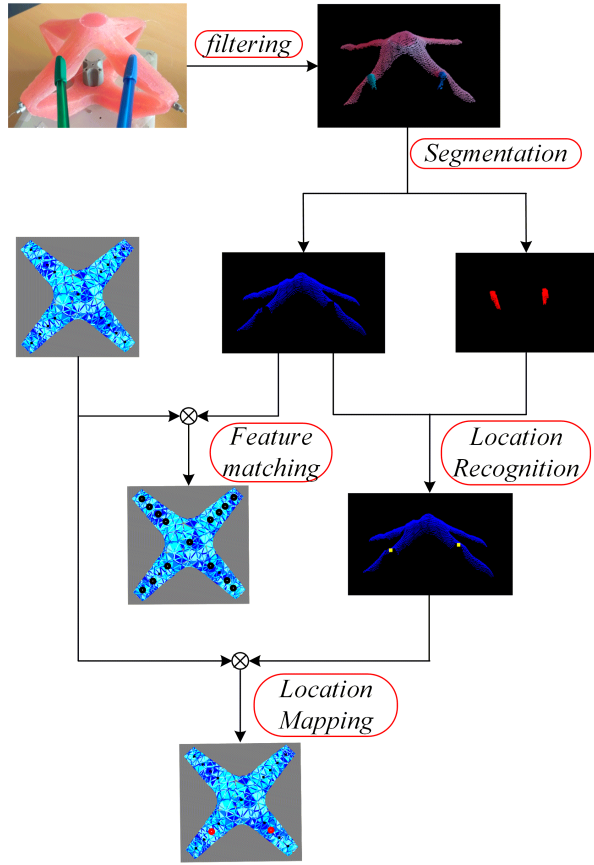


Fig. 2. Location recognition and feature points matching. The image processing method consists of five steps: filtering, segmentation, location recognition, location mapping and feature points matching.

An acquired point cloud is then filtered through PassThrough, VoxelGrid filters and RadiusOutlier removal, to remove points whose values fall outside a given interval along a specified dimension, to downsample the point cloud and to remove outliers from noisy data, respectively.

2) *Segmentation*: We assume that the color of the robot is homogeneous and distinct from the ones of the surrounding objects. Using a color-based region growing segmentation technique, the filtered point cloud is segmented into two parts: the point cloud of the soft robot, defined as \mathbf{X}_r (shown in blue in Fig. 2) and the point cloud of the surrounding objects, defined as \mathbf{X}_o (shown in red in Fig. 2).

3) *Contact location detection*: The contact locations between the soft robot and the surrounding objects are estimated by processing both point clouds. The algorithm is summarized in Algorithm 2 and can be described as follows.

The contact points between the robot and the colliding objects are usually difficult to be detected because they can be occluded by the contact operator. When a contact occurs, some points in \mathbf{X}_o get close to \mathbf{X}_r . For each point \mathbf{x}_o in \mathbf{X}_o , a nearest neighbors search, using kd trees, is then performed among \mathbf{X}_r . If the number of neighbors is larger than a defined threshold number n_{neigh} , we regard \mathbf{x}_o as a contact point.

When all the contact points are collected, they are clustered into several groups based on euclidean cluster extraction. The

average position \mathbf{x}_d in each group is computed as the detected location of an external force.

Algorithm 2 Contact location detection

1. **Define** a constant search radius r_{search} and a threshold number of neighbors n_{neigh}
 2. **If** objects are found after segmentation
 - For** each point in contact point cloud
 - Search neighbors (r_{search}) in robot point cloud
 - If** number of neighbors larger than n_{neigh}
 - Feature points detected
 - End**
 - Separate the detected points into groups
 - For** each group
 - Compute the average position
 - End**
-

4) *Location mapping*: The detected locations are mapped to the surface of the FE model where multiple feature points $\mathbf{X}_e = \{\mathbf{x}_{e,k}\}_{k=0}^P \subset \mathbf{X}_{fe}$ with \mathbf{X}_{fe} the set of vertices of the FE model are predefined and evenly distributed on the surface, as shown in Fig. 2. These feature points are employed to approximate the surface of the robot. For each detected location \mathbf{x}_d , we find the feature point \mathbf{x}_{e,k_d} on the FE model which is the closest to \mathbf{x}_d . Force estimation will then be performed on the obtained subset of contact feature points $\{\mathbf{x}_{e,k_d}\}$ on which the detected contact locations are mapped.

5) *Feature points matching*: External force sensing requires the computation of the displacements of the feature points \mathbf{X}_e predefined on the surface of both the soft robot and its FE model. Since these predefined feature points \mathbf{X}_e on the FE model are not recognizable like fiducial markers, a matching technique between \mathbf{X}_e and the observed point cloud of the robot \mathbf{X}_r is proposed, as described in Algorithm 3 and as follows.

For each feature point \mathbf{x}_e , we perform matching by looking for a correspondence along its normal direction \mathbf{n}_e which is computed using the surface of FE model (see Fig. 3). Then we find the point \mathbf{x}_c in \mathbf{X}_r which has the minimum distance d_{mini} to the normal line. Each feature point \mathbf{x}_e and the found point \mathbf{x}_c are regarded as a correspondence, for which the displacement of \mathbf{x}_e can be computed. We define $\mathbf{X}_c = \{\mathbf{x}_{c,k}\}_{k=0}^P \subset \mathbf{X}_r$ as the set of correspondences.

In real world application, some feature points may not be available because of occlusions. During matching, the hidden feature points on the FE model need to be detected. Consequently, we predefine a threshold distance d_m to discard potential outlier feature points. For each \mathbf{x}_e , if the distance to its corresponding point in \mathbf{X}_r is larger than the threshold, \mathbf{x}_e is not available and its displacement will not be considered for the intensity computation described in the next subsection.

The matching process should also avoid the situation for which the point cloud is not well segmented, e.g. objects with colors similar to the ones of the robot. A threshold distance d_l is predefined to detect the matches which fall into this situation. If the distance between a feature point \mathbf{x}_e and its match \mathbf{x}_c is larger than d_l , the match is considered as an outlier and \mathbf{x}_e is discarded for force computation.

Algorithm 3 Feature point matching

```

1. Define feature points on the surface of FE model and
   two threshold distance  $d_m$  and  $d_l$ 
2. For each feature point  $\mathbf{x}_e$ 
   Compute the normal direction
   For each point in robot point cloud
     Compute distance between  $\mathbf{x}_e$  to normal line
   End
   Compute the minimum distance  $d_{mini}$ 
   If  $d_{mini} > d_m$ , the feature point is hidden
   Else possible match  $\mathbf{x}_c$  obtained
   Compute distance  $d$  between  $\mathbf{x}_e$  to  $\mathbf{x}_c$ 
   If  $d < d_l$ 
     match  $\mathbf{x}_c$  obtained
   End

```

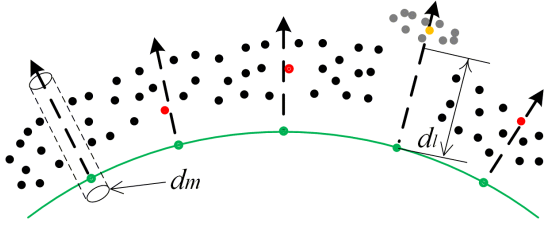


Fig. 3. Matching process of the feature points along the normal direction. The black, green, gray, orange and red points are respectively the robot point cloud, effector point cloud, outliers after segmentation, matches in the outliers and the matches for effectors. d_m and d_l are the predefined threshold values to remove imprecise matches in the segmented point cloud.

B. Intensity computation

The external forces at the found contact feature points $\{\mathbf{x}_{e,k_d}\}$ are computed so that on the FE model deforms towards the real robot and its point cloud. By relying on the methodology described in section III.A, we map the quasi-static equation (3) to the motion space along the normal of the robot surface.

$$\delta_e^N = \mathbf{W}_{ea}^N \lambda_a + \mathbf{W}_{ef}^N \lambda_f + \delta_e^{free,N} \quad (10)$$

where $\delta_e^N = [\mathbf{x}_{e,0}^T \mathbf{n}_{e,0}^T \cdots \mathbf{x}_{e,P}^T \mathbf{n}_{e,P}^T]$ and $\delta_e^{free,N} = [\mathbf{x}_{e,0}^{free} \mathbf{n}_{e,0}^T \cdots \mathbf{x}_{e,P}^{free} \mathbf{n}_{e,P}^T]$.

It is indeed easier to capture this direction of motion than other directions, due to ambiguities in the nearest neighbors searches along the tangential plane. For the same number of effectors, this strategy can reduce the constraint size and further reduce the computation time.

The compliance matrices \mathbf{W}_{ea}^N and \mathbf{W}_{ef}^N can be obtained using the FE method. Based on the approach described in Fig. 1, the intensity of the external forces at the contact feature points $\{\mathbf{x}_{e,k_d}\}$ can be computed by solving an optimization problem $\lambda = \arg \min \Gamma(\lambda)$ where $\Gamma(\lambda)$ is the objective function, with respect to λ_f , contributing on the contact feature points:

$$\Gamma(\lambda) = \frac{1}{2} \|\mathbf{W}_{el}(\mathbf{x}) \lambda_I + \delta_e^{free,N} - \delta_c^N\|^2 \quad (11)$$

where $\mathbf{W}_{el} = [\mathbf{W}_{ea}^N \mathbf{W}_{ef}^N]$, $\lambda_I = [\lambda_a \lambda_f]$ and $\delta_c^N = [\mathbf{x}_{c,0}^T \mathbf{n}_{c,0}^T \cdots \mathbf{x}_{c,P}^T \mathbf{n}_{c,P}^T]$. The computed external forces λ_f and λ_a are employed to update the FE model to reach the same configuration of the robot observed by the camera.

In the case of a constrained optimization, a numerical optimization algorithm is necessary to be employed. In order to match the standard quadratic programming (QP) formulation for the usage of software packages [25], (11) can be converted to

$$\min_{\lambda} \frac{1}{2} \lambda^T \mathbf{Q}_s \lambda + \mathbf{c}_s^T \lambda \quad (12)$$

$$\text{s.t. } \lambda_a = \lambda_a^r \quad (13)$$

$$\text{or } \mathbf{W}_a^N \lambda + \delta_a^{free,N} = \delta_a^{r,N} \quad (14)$$

where $\mathbf{Q}_s = \mathbf{W}_{el}^T \mathbf{W}_{el}$ and $\mathbf{c}_s = \mathbf{W}_{el}^T (\delta_e^{free,N} - \delta_c^N)$. λ_a^r and $\delta_a^{r,N}$ are respectively the actuator force and actuator displacement sent to both the soft robot and its FE model. The control inputs are considered for the computation by the constraints (13) and (14). (13) and (14) are employed respectively when the actuator force and the actuator displacement are chosen as the control input.

Remark 1. The dimension of \mathbf{Q}_s is equal to the number of external forces and actuators so that the computation cost for the QP optimization (12) remains almost constant, even if a larger number of feature points is employed for the computation.

VI. EXPERIMENT RESULTS

The strategy proposed in this paper is validated by some experiments described in this section. In the first subsection, the experimental setup is introduced. Then the performances of the calibration and force sensing methods are shown in the following two subsections.

A. Experimental setup

As shown in Fig. 4, the experimental setup consists of three components: a soft parallel robot, a real-time FE model, and an RGB-D camera. The soft robot has four legs and is actuated by cables on each leg so that it is difficult to get an analytical kinematic model. It is made of silicone with the Young modulus 180kPa. The FE model of the soft robot is built with 4884 elements and is simulated in the SOFA framework [26], which is a real-time FEM simulation software. The methods for FEM simulation and soft robot modeling are described in details in [24]. The configuration of the soft robot is updated with the frequency of 22 Hz (5 markers) for calibration and 7Hz (51 feature points) for external force sensing. The *Realsense* depth camera D415 by Intel company is employed to capture the scene. The frequency of the image processing is 10Hz for force sensing.

B. Calibration performances

The calibration technique is validated by both simulation and real experiments. The variables to be calibrated in this paper are \mathbf{R} , \mathbf{T} and λ_a as shown in (7).

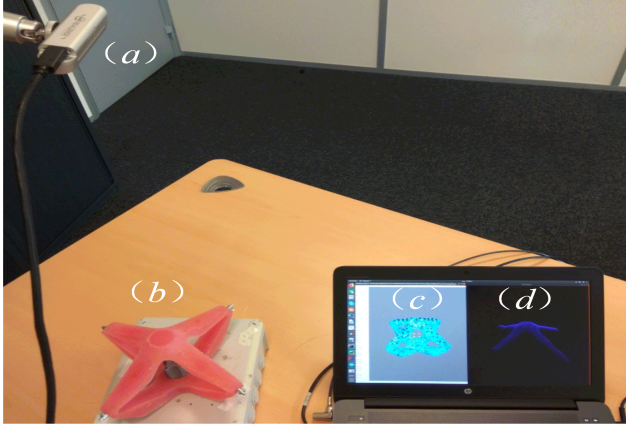


Fig. 4. Experimental setup for calibration and external force sensing of soft robot. The setup consists of an RGB-D camera (a) which generates the point cloud of robot surface (d), a soft parallel robot (b), a real-time FE model (c).

1) *Validation by simulation:* As shown in Fig. 5, two cases with different number and distribution of markers are considered for the validation. In the first case, five markers (the white dots in Fig. 5) are defined. Nine feature points (both black and white dots in Fig. 5) are employed in the second case. The objective of the calibration technique is to actuate the initial configuration (Fig. 5 (a)) with respect to $oxyz$ in order to reach the desired one (Fig. 5 (b)) with respect to $pxyz$. The calibrated configuration is shown in Fig. 5 (c) with respect to $oxyz$.

The actual values of all variables for both cases are shown in Table. I. The calibrated values and the residual errors of the objective function (7) are listed in Table. I. Being validated by simulation, the proposed calibration strategy can provide calibrated parameters with a high accuracy.

The residual errors with respect to the number of iterations is shown in Fig. 6. The calibration with nine markers has a faster convergence and a higher accuracy so that we use it for the real world experiment.

TABLE I

CALIBRATION RESULTS USING SIMULATION. THE UNIT OF λ IS NEWTON. THE UNIT OF BOTH \mathbf{T} AND THE RESIDUAL ERROR IS MILLIMETER. AV: ACTUAL VALUE; RE: RESIDUAL ERROR; (I): THE VALIDATION WITH FIVE MARKERS; (II) THE VALIDATION WITH NINE MARKERS.

Items	\mathbf{R}			\mathbf{T}	λ
AV	-0.147	0.941	0.305	-264.956	1.200
	-0.485	0.200	-0.851	263.885	1.000
	-0.862	-0.273	0.427	12.767	1.000
					1.200
RE(I) 0.025	-0.147	0.941	0.305	-264.965	1.199
	-0.484	0.200	-0.852	263.913	1.000
	-0.862	-0.273	0.426	12.947	0.999
					1.200
RE(II) 0.024	-0.147	0.941	0.305	-264.938	1.200
	-0.485	0.200	-0.851	263.899	1.001
	-0.862	-0.273	0.427	12.799	1.001
					1.201

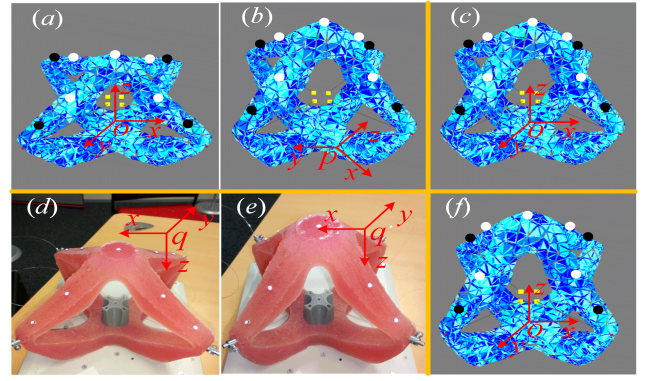


Fig. 5. Validation design for the calibration by simulation ((a), (b) and (c)) and real experiment ((d), (e) and (f)). (a), (b) and (c) are respectively the initial, the desired and the calibrated (based on (b)) configurations of the FE model with respect to $oxyz$, $pxyz$ and $oxyz$. (d) and (e) are respectively the initial and the desired configurations of soft robot with respect to $qxyz$. (f) is the calibrated (based on (e)) configurations of the FE model with respect to $oxyz$. The spots (white and black) are the markers on the FE model.

TABLE II

CALIBRATION RESULTS USING THE REAL WORLD EXPERIMENT. THE UNIT OF \mathbf{T} , δ_a , AND THE RESIDUAL ERROR IS MILLIMETER. AV: ACTUAL VALUE; RE: RESIDUAL ERROR; (II) THE VALIDATION WITH NINE MARKERS.

Items	\mathbf{R}			\mathbf{T}	δ_a
AV	-0.732	0.679	-0.059	17.480	3.947
	0.681	0.729	-0.064	12.598	5.176
	0.000	-0.087	-0.996	566.918	7.920
					7.644
RE(II) 7.47	-0.736	0.674	-0.060	17.338	3.249
	0.677	0.735	-0.039	-1.566	5.065
	0.018	-0.069	-0.997	570.349	10.921
					8.904

2) *Validation by real world experiment:* Instead of using the direct FE model, the real soft robot (Fig.5 (d), (e)) is employed for validation. Here, we show the validation results using nine markers which are selected and manually measured from the point cloud. Fig. 5 (a) is also the initial configuration of the FE model. Fig.5 (e) and (f) are respectively the configuration of the soft robot with respect to $qxyz$ and the calibrated configuration of the FE model with respect to $oxyz$.

Table II shows the residual error for the real world experiment. The cable displacement δ_a (computed by (4) with $\lambda_f = 0$) is chosen as the variable for the validation. The initial configuration of the soft robot (Fig.5 (d)) is employed to get the actual value of \mathbf{R} , \mathbf{T} . The actual value $\delta_a = \delta_a^{initial} + \Delta\delta_a$ where $\delta_a^{initial}$ is computed by the calibration based on the initial configuration of the soft robot (Fig.5 (d)) and $\Delta\delta_a$ is randomly set by users. Actuated with δ_a , the configuration of soft robot is shown in Fig. 5 (e).

As shown in Fig. 6, the residual error can be reduced to 7.47 mm after ten iterations in our experiment. The residual error is larger than errors in simulation. This is due to errors in the FE model, in the image processing step and in the correspondences between the artificial markers and the feature points defined on the FE model.

The calibration method has several limitations: (1) A

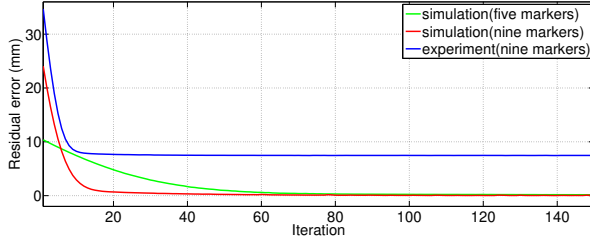


Fig. 6. Residual error of the validation with respect to the number of iterations by simulation and real world experiment.

limited number of feature points should be defined for the FE model to balance the computation time and external force sensing accuracy; (2) The material parameters needs to be calibrated beforehand; (3) The manual selection of the feature points in the point cloud of the soft robot results in errors to find the corresponding points on the FE model.

C. External force sensing performances

Given the accurate initial configuration of the FE model provided by the calibration step, the external force sensing can be handled. In this subsection, we show the performances of image processing, external force computation and the compound of location detection and intensity computation.

1) *Performance of image processing:* The image processing step in this paper is based on some algorithms provided in the Point Cloud Library (PCL)[27]. The objective is to segment the point cloud of the robot and the one of its surrounding environment. The color-based region growing segmentation technique is employed, with the assumption that the color of the robot is homogeneous and is different with respect to its surrounding. As shown in Fig. 7, the contact points can be detected successfully with the blue points being the surface of the robot and the yellow points being the contacts location.

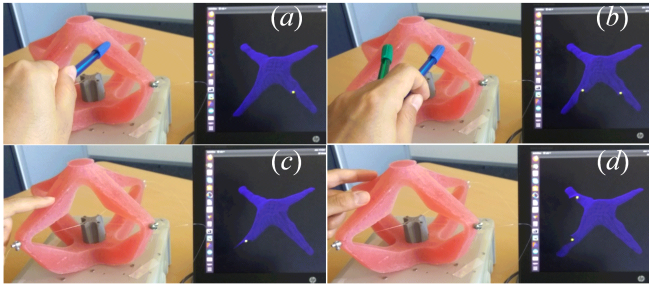


Fig. 7. Screenshot of image processing results on the location of external forces. The yellow points are the detected locations of external forces. (a) and (c) are the situation of one external forces. (b) and (d) correspond to the detection of two external forces.

2) *Performance of external force computation:* Multiple three-dimensional external forces can be computed and tracked using the proposed methods. But in this subsection, we focus on one-dimensional force computation. As shown in Fig. 8, the external force generated by a weight in z-direction which is along the direction of gravity is applied on the tip of the robot. The control inputs are also considered with

constant cable displacements. Based on the inverse FE model, the intensity of the weight is computed and is listed in Table. III for different weight. The Young's Modulus is set to be 180 kPa which is calibrated based on the relation between weight and deformation. From Table III, the computed intensity can reach a higher level of accuracy where the average error for six weights is around 4%.

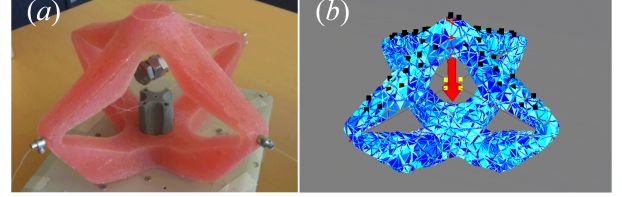


Fig. 8. Experiment setup for the validation of intensity computation. (a) the soft robot with a weight hanging on the tip; (b) the FE model with feature points and computed force

TABLE III
INTENSITY COMPUTATION. THE FORCE UNIT IS *mN*.

Actual force	260	460	660	760	860	1060
Computed forces	230	434	634	756	873	1045
Absolute error	30	26	26	4	7	15
Computation error	11.5%	5.7%	3.9%	0.5%	1.5%	1.4%

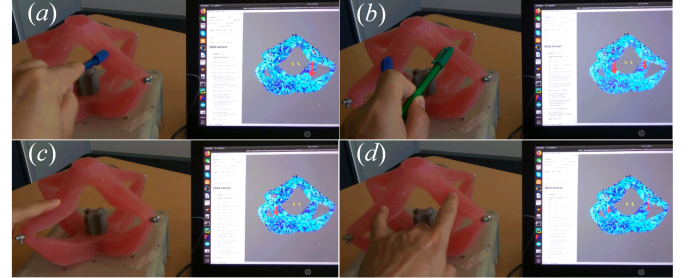


Fig. 9. Screenshot of external force sensing. The robot has four cables with constant length for the experiment. (a) and (c) show one external force on the actuated soft robot. (b) and (d) show case with two external forces.

3) *External force sensing:* In Fig. 9, the screenshots of the external force sensing system is shown with the integration of location estimation and intensity computation. In the experiment, we consider two cases: the soft robot with one external force ((a) and (c)) and two external forces ((b) and (d)). 51 feature points are pre-defined on the surface of FE model and are regarded as the possible locations of external forces. When the locations of the external forces are detected based on the image processing of the point cloud, the correct locations on the FE model are selected. Then the intensities are computed by solving the optimization problem described in section V.B so that the FE model can be actuated to align to the corresponding point cloud.

The limitations of the marker-free force sensing approach are: (1) We employ a color-based image segmentation technique for the point cloud which reduces the generalization and is sensitive to the color contrast between the considered areas; (2) The external forces are assumed to be point-wise forces

which are applied on the surface of the soft robot. Therefore, the estimation accuracy reduces if the contact area between the robot and its environment is larger; (3) The implementation needs the visibility of the contact points by the camera.

VII. CONCLUSION

This paper proposes a strategy for calibration and external force sensing for soft robots. By choosing several corresponding feature points on the surface of a soft robot and its FE model, we can calibrate the sensor-robot system and the initial actuator input. The calibration is based on the numerical model of the soft robot and can be employed for robots with complex structures. The marker-free external force sensing technique relies on matching the point cloud provided by an RGB-D camera and predefined feature points on the surface of the FE model. Using the external force sensing strategy, both the location and the intensity of external forces on the entire surface can be estimated. This method could be used in more general computer vision applications involving deformable objects and requiring external force estimation (for instance for robotic manipulation, surgery, augmented reality and motion capture problems).

In our future work, we would like to extend the strategy for the calibration of geometric and material parameters. In order to improve the performances of external force sensing, we would like to employ a more robust approach for point cloud segmentation and to use a multi-camera system to capture the whole shape of the soft robot and its contacts. GPU programming would be employed for the simulation and image processing steps to increase the response speed of the force sensing algorithm. We also plan to extend the system to the hybrid position/force control based on the FE model. Besides, we would like to evaluate the proposed techniques via concrete real world applications.

REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, p. 467, 2015.
- [2] K. Xu and N. Simaan, "An investigation of the intrinsic force sensing capabilities of continuum robots," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 576–587, 2008.
- [3] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2018.
- [4] Z. Zhang, J. Dequidt, A. Kruszewski, F. Largilliere, and C. Duriez, "Kinematic modeling and observer based control of soft robot using real-time finite element method," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 5509–5514, IEEE, 2016.
- [5] Z. Zhang, T. M. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual servoing control of soft robots based on finite element model," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, IEEE, 2017.
- [6] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018.
- [7] A. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, "An overview of robot calibration," *Information Technology Journal*, vol. 3, no. 1, pp. 74–78, 2004.
- [8] M. Shah, R. D. Eastman, and T. Hong, "An overview of robot-sensor calibration methods for evaluation of perception systems," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pp. 15–20, ACM, 2012.
- [9] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. M. Pathak, "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME transactions on mechatronics*, vol. 20, no. 2, pp. 663–674, 2015.
- [10] N. Tan, X. Gu, and H. Ren, "Simultaneous robot-world, sensor-tip, and kinematics calibration of an underactuated robotic hand with soft fingers," *IEEE Access*, 2017.
- [11] F. Thérien and J.-S. Plante, "Design and calibration of a soft multiple degree of freedom motion sensor system based on dielectric elastomers," *Soft Robotics*, vol. 3, no. 2, pp. 45–53, 2016.
- [12] B. Wang, L. Wu, K. Yin, U. Ascher, L. Liu, and H. Huang, "Deformation capture and modeling of soft objects," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 94, 2015.
- [13] A. Petit, F. Ficuciello, G. A. Fontanelli, L. Villani, and B. Siciliano, "Using physical modeling and rgb-d registration for contact force sensing on deformable objects," in *ICINCO 2017-14th International Conference on Informatics in Control, Automation and Robotics*, vol. 2, pp. 24–33, ScitePress; Springer, 2017.
- [14] P. Puangmali, H. Liu, L. D. Seneviratne, P. Dasgupta, and K. Althoefer, "Miniature 3-axis distal force sensor for minimally invasive surgical palpation," *IEEE/ASME Transactions On Mechatronics*, vol. 17, no. 4, pp. 646–656, 2012.
- [15] U. Kim, D.-H. Lee, W. J. Yoon, B. Hannaford, and H. R. Choi, "Force sensor integrated surgical forceps for minimally invasive robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1214–1224, 2015.
- [16] R. Xu, A. Yurkewich, and R. V. Patel, "Curvature, torsion, and force sensing in continuum robots using helically wrapped fbg sensors," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1052–1059, 2016.
- [17] F. Khan, R. J. Roesthuis, and S. Misra, "Force sensing in continuum manipulators using fiber bragg grating sensors," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017*, 2017.
- [18] J. Back, T. Manwell, R. Karim, K. Rhode, K. Althoefer, and H. Liu, "Catheter contact force estimation from shape detection using a real-time cosserat rod model," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2037–2042, IEEE, 2015.
- [19] S. Hasanzadeh and F. Janabi-Sharifi, "Model-based force estimation for intracardiac catheters," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 154–162, 2016.
- [20] M. Khoshnam, A. C. Skanes, and R. V. Patel, "Modeling and estimation of tip contact force for steerable ablation catheters," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 5, pp. 1404–1415, 2015.
- [21] A. Bajo and N. Simaan, "Kinematics-based detection and localization of contacts along multisegment continuum robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 291–302, 2012.
- [22] E. Magrini, F. Flacco, and A. De Luca, "Estimation of contact forces using a virtual force sensor," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2126–2133, IEEE, 2014.
- [23] T.-H. Pham, A. Kheddar, A. Qammar, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2810–2819, 2015.
- [24] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [25] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [26] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtcuisse, G. Bousquet, I. Peterlik, *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pp. 283–321, Springer, 2012.
- [27] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011.